# PORTAL
## THE ACM DIGITAL LIBRARY

| Try the *new* Portal design |
| --- |
| Give us your opinion after using it. |

## Search Results

Search Results for: **[partition<AND>((profile<AND>(((code restructuring)) )) )]**
Found **7** of **127,132 searched.**

## Search within Results

[                                                    ]  GO    > Advanced Search   > Search Help/Tips

---

**Sort by:    Title    Publication    Publication Date    Score    📖 Binder**

---

**Results 1 - 7 of 7        short listing**

---

**1** Research track: Improving spatial locality of programs via data mining          77%
Karlton Sequeira , Mohammed Zaki , Boleslaw Szymanski , Christopher Carothers
**Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining** August 2003
> In most computer systems, page fault rate is currently minimized by generic page replacement algorithms which try to model the temporal locality inherent in programs. In this paper, we propose two algorithms, one greedy and the other stochastic, designed for program specific code restructuring as a means of increasing spatial locality within a program. Both algorithms effectively decrease average working set size and hence the page fault rate. Our methods are more effective than traditional appr ...


**2** Object equality profiling          77%
Darko Marinov , Robert O'Callahan
**ACM SIGPLAN Notices , Proceedings of the 18th ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications** October 2003
Volume 38 Issue 11
> We present *Object Equality Profiling* (OEP), a new technique for helping programmers discover optimization opportunities in programs. OEP discovers opportunities for replacing a set of equivalent object instances with a single representative object. Such a set represents an opportunity for automatically or manually applying optimizations such as hash consing, heap compression, lazy allocation, object caching, invariant hoisting, and more. To evaluate OEP, we implemented a tool to help prog ...


**3** A comparison of automatic parallelization tools/compilers on the SGI origin          77%
2000
Michael Frumkin , Michelle Hribar , Haoqiang Jin , Abdul Waheed , Jerry Yan
**Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)** November 1998
> Porting applications to new high performance parallel and distributed computing platforms is a challenging task. Since writing parallel code by hand is time consuming and costly, porting codes would ideally be automated by using some parallelization tools and compilers. In this paper, we compare the performance of three parallelization tools and compilers based on the

NAS Parallel Benchmark and a CFD application, ARC3D, on the SGI Origin2000 multiprocessor. The tools and compilers compared inclu ...

## 4 Load-reuse analysis: design and evaluation

77%

Rastislav Bodík , Rajiv Gupta , Mary Lou Soffa
**ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation** May 1999
Volume 34 Issue 5

Load-reuse analysis finds instructions that repeatedly access the same memory location. This location can be promoted to a register, eliminating redundant loads by reusing the results of prior memory accesses. This paper develops a load-reuse analysis and designs a method for evaluating its precision.In designing the analysis, we aspire for *completeness*---the goal of exposing all reuse that can be harvested by a subsequent program transformation. For register promotion, a suitable transfo ...

## 5 Register promotion by sparse partial redundancy elimination of loads and stores

77%

Raymond Lo , Fred Chow , Robert Kennedy , Shin-Ming Liu , Peng Tu
**ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation** May 1998
Volume 33 Issue 5

An algorithm for register promotion is presented based on the observation that the circumstances for promoting a memory location's value to register coincide with situations where the program exhibits partial redundancy between accesses to the memory location. The recent SSAPRE algorithm for eliminating partial redundancy using a sparse SSA representation forms the foundation for the present algorithm to eliminate redundancy among memory accesses, enabling us to achieve both computational and li ...

## 6 Complete removal of redundant expressions

77%

Rastislav Bodík , Rajiv Gupta , Mary Lou Soffa
**ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation** May 1998
Volume 33 Issue 5

Partial redundancy elimination (PRE), the most important component of global optimizers, generalizes the removal of common subexpressions and loop-invariant computations. Because existing PRE implementations are based on *code motion*, they fail to completely remove the redundancies. In fact, we observed that 73% of loop-invariant statements cannot be eliminated from loops by code motion alone. In dynamic terms, traditional PRE eliminates only half of redundancies that are strictly partial. ...

## 7 The design of a new frontal code for solving sparse, unsymmetric systems

77%

I. S. Duff , J. A. Scott
**ACM Transactions on Mathematical Software (TOMS)** March 1996
Volume 22 Issue 1

We describe the design, implementation, and performance of a frontal code for the solution of large, sparse, unsymmetric systems of linear equations. The resulting software package, MA42, is included in Release 11 of the Harwell Subroutine Library and is intended to supersede the earlier MA32 package. We discuss in detail the extensive use of higher-level BLAS kernels within MA42 and illustrate the performance on a range of practical problems on a CRAY Y-MP, an IBM 3090, and an IBM RISC Sys ...

**Results 1 - 7 of 7     short listing**